NASA/Army

Systems and Software Engineering Forum

Improving Software Engineering on NASA Projects

May 11, 2010

Tim Crumbley & John C. Kelly

Office of Chief Engineer

NASA Software Engineering Initiative

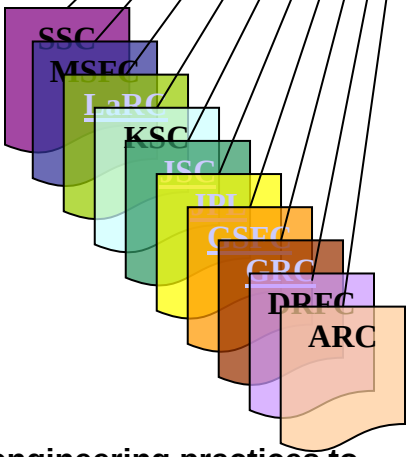
NASA's DOD & Industry Interface

Software Engineering Initiative

Special NASA Software Engineering Studies

Flight Software Complexity
Fault Management Workshop
Programmable Logic Devices (Complex Electronics)

1. Center SEPGs
Process Improvements
SW Metrics



2. Software Working Group

3. SW Policy
NASA NPD 2820.1 & NPR 7150.2

NASA SW Steering Board

CMMI Steering Group

4. Training, CMMI Appraisals & Career Development

5. SW Technology Infusion & Journal

6. Engineering, Assurance, & Safety Collaboration
NASA SW Process Asset Library
NASA SW Inventory
NASA Engineering Network (Software)

Objective:
"…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

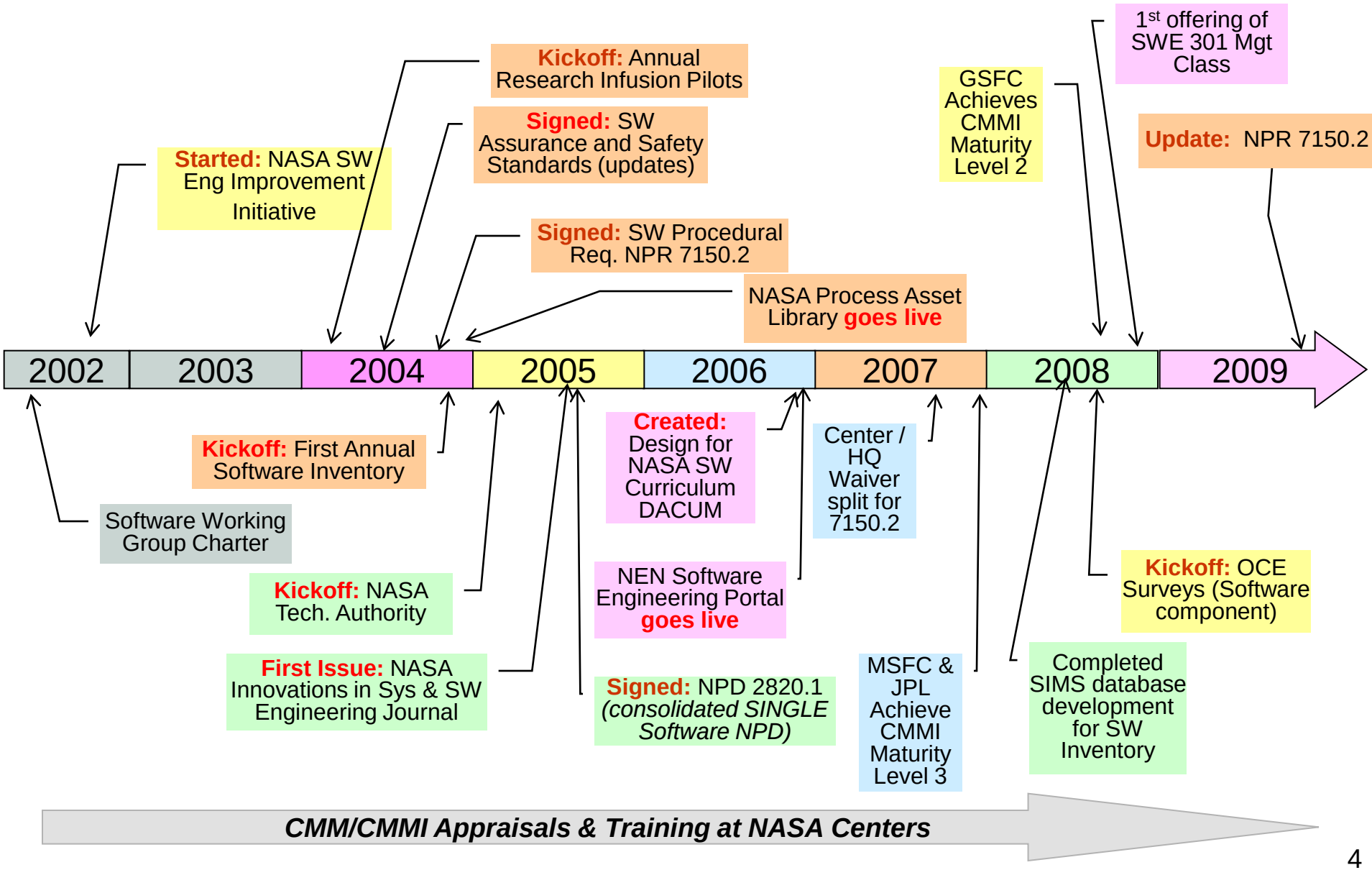
Sponsor: NASA Office of Chief Engineer

FY 2010 Software Improvement Initiative Plans

	Policy & Procedural Requirements	Processes	Technology
Ongoing	• NPD 7120.4* - ***completed*** • NPR 7150.2A, SW Engineering Requirements update - ***completed*** • OCE Survey* (10 Centers + HQ)	• CMMI Appraisals • NASA & Center Process Asset Libraries (PALs) • SW Measurement	• Tool Shed • Sys & SW Journal • *Reviewers* and *Rep*. to OSMA's SW Assurance Research Program (SARP)*
New for 2010	• SW Engr. Handbook (Electronic) • Center Compliance with new NPR 7150.2A (Phase 1) • OSMA's update to NASA Safety and Assurance standards • *Representative* to help develop new Programmable Logic Devices Policy/Std/HB*	• Center processes updated for consistency with new NPR 7150.2A	• Update SW Technology Strategy for 2011 and beyond • *Interface* to SW Architecture Review Board effort (NESC)* • *Interface* to Multi-Core Flight computing* • *Interface* to SW Engineering Research Center (SERC) • *Interface* to NASA Aviation Safety Program*
Crosscutting	• Center SW Improvement Plans • Training (including NPR 7150.2A Classroom & NASA SATERN) • NASA Engineering Network*, Software.nasa.gov • Software Inventory, SIMS Tool, Analysis & Suggestions for projects to receive IV&V • SWG F2Fs, Leads Meeting, & Telecons • Communications / Exchanges (CMMI Steering Group, v1.3 CCB, TIMs, etc.) • *Interface* to Systems Engineering Working Group		

**** Software Engineering portions or contributions***

Timeline 2002 – 2009
NASA Software Engineering Initiative



Top Software Issues from NASA Centers 2007

1. Software Requirements
2. Internal NASA-wide requirements (NPD, NPR, & Standards)
3. Training & Skill Development
4. Complex Electronics, FPGA, PLD, etc. (blurring of H/W – S/W boundary)
5. Insight/Oversight of Contractor SW development
6. Tools
7. Empowerment of program/project SW personnel
8. Metrics/Measurement
9. COTS -Impacts of maintaining COTS and technologies for long-term systems and missions
10. Cost estimation - need a standard approach

Top Software Issues from NASA Centers 2010

1. Internal NASA-wide NASA Procedural Requirements & Standards (including Software Classification) (2)
2. Cost Estimation (10)
3. Software Workforce Level (New)
4. Systems Engineering / Software Engineering Interfaces (New)
5. Small Project Implementations (cross cutting) (New)
6. Empowerment of Software Engineering Personnel (7)
7. Software Requirements (1)
8. Complex Electronics (4)
9. Training & Skill Development (3)
10. Insufficient attention to Software on Contracts (New)

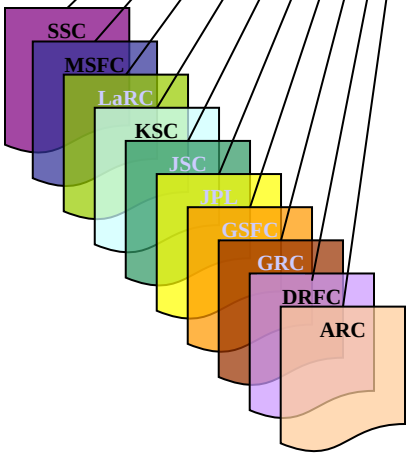
Software Engineering Initiative

Special NASA Software Engineering Studies

Flight Software Complexity
Fault Management Workshop
Programmable Logic Devices (Complex Electronics)

Software Engineering Initiative

1. Center SEPGs
Process Improvements
SW Metrics



2. Software Working Group

3. SW Policy
NASA NPD 2820.1 & NPR 7150.2

NASA SW Steering Board

CMMI Steering Group

4. Training, CMMI Appraisals & Career Development

5. SW Technology Infusion & Journal

6. Engineering, Assurance, & Safety Collaboration
NASA SW Process Asset Library
NASA SW Inventory
NASA Engineering Network (Software)

Objective:
"…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer

GODDARD
SPACE FLIGHT CENTER

GSFC Software Process Improvement

If the Process Works...Improve It.

+ GSFC SW IMPROVEMENT | + PROCESS ASSETS LIBRARY | + TRAINING | TOOLS | + MEASURES | + LESSONS LEARNED

SPI Initiative
The Navigation Bar
Sit
+ S
+ P
Co

Training to ensure project engineers have necessary skills

Local tools to support projects

News and ming Events

Upda
"Bran
"Fund
"Phys
"Baseline Audit Checklist"

Link to New Assets --->

Welcome

The goal of GSFC Software Process Improvement is to establish and continuously improve system and software processes and products by providing the necessary supporting infrastructure, such as tools, templates, measurements support, and lessons learned. The objectives are to:

- Improve the quality, reliability, and safety of our products through the integration of sound system and software engineering principles and standards, so that our customers receive highly effective and reliable products that fulfill their scientific and

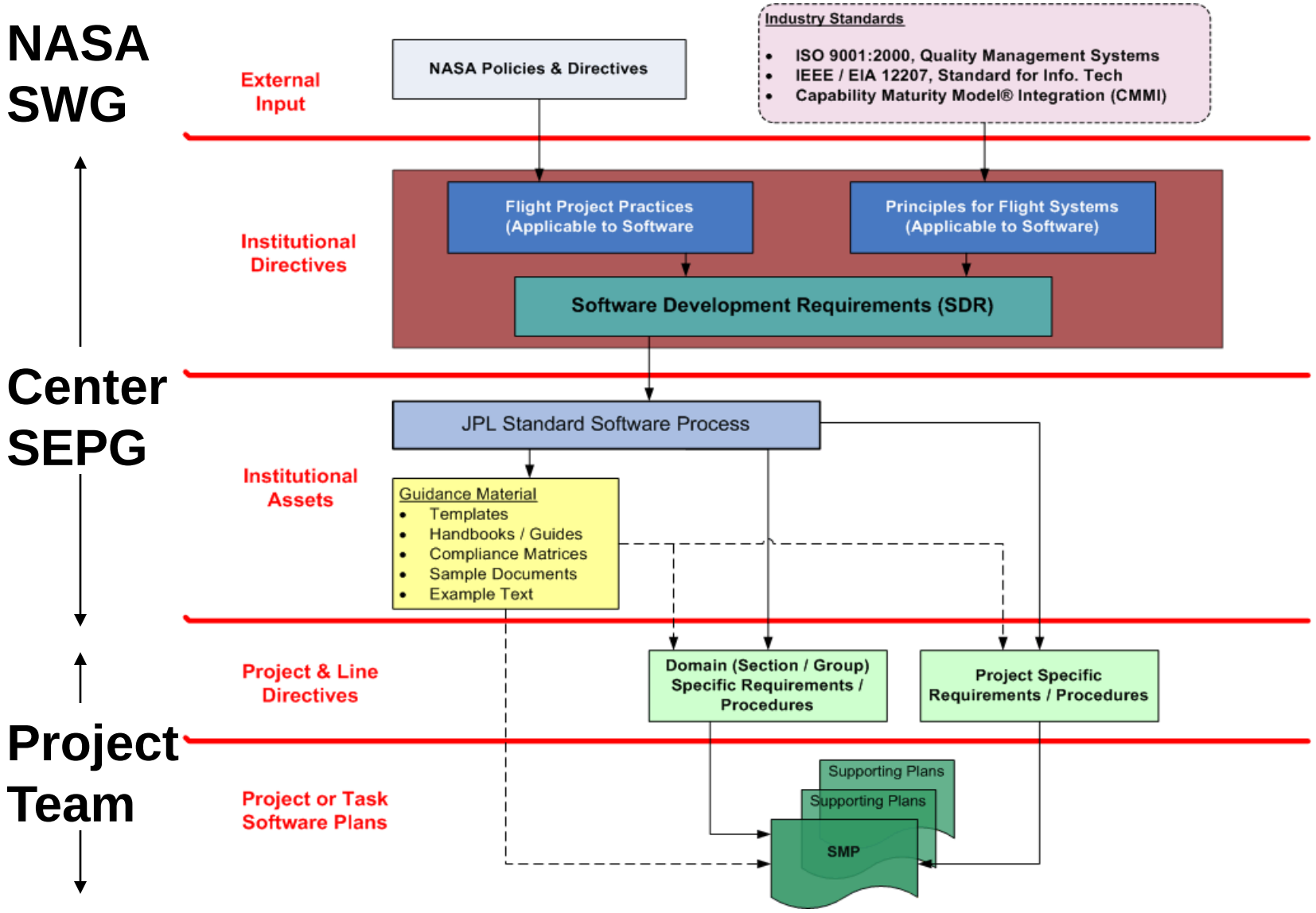
Conferences

Carnegie Mellon Software Engineering Institute (SEI) Software Engineering Process Group North America conference (SEPG '09)
March 23-26, 2009
San Jose, California, USA

The 21st Annual Systems & Software Technology Conference (STC 2009)
April 20-23, 2009
Salt Lake City, Utah, USA

Southeastern Software & Systems Engineering

Requirements Flow



NASA CMMI Summary

Completed CMMI Appraisals

Center	Rating	Date	# Projects	Type	Project size
LaRC- ASDC	PP(CL3), CM(CL1)	17-Nov-06	1	Data Center Support	85
MSFC	ML3	24-Apr-07	3	Development	57,4,2 = 63
JPL	ML3	27-Sep-07	7	Dev & Maintenance	1,8,11,17,16,1,30 = 84
GSFC	ML2 + RSKM(2)	16-May-08	4	Dev & Maintenance	25,2,18,8 = 53
LaRC- FSSB	ML2 + CL3	3-Oct-08	3	Services	1,1,3 = 5
LaRC- SDAB	PP(CL3), REQM(CL3), CM(CL3), MA(CL3)	13-Mar-09	4	Development	1,5,10,5 = 21
JSC	ML2	2-Apr-09	4	Development	7,6,45,2 = 60
KSC	ML2	18-Sep-09	1	Development	225

Scheduled CMMI Appraisals in FY10

SCAMPI A

Center	Month
MSFC	April
ARC	May
MSFC	June
GRC	August
JPL	September

SCAMPI B

Center	Month
JPL	October
GSFC	June

SCAMPI C

Center	Month
GSFC	February
JSC	April

Software Engineering Initiative

Special NASA Software Engineering Studies

Flight Software Complexity
Fault Management Workshop
Programmable Logic Devices (Complex Electronics)

Software Engineering Initiative

1. Center SEPGs
Process Improvements
SW Metrics

SSC
MSFC
LaRC
KSC
JSC
JPL
GSFC
GRC
DRFC
ARC

2. Software Working Group

3. SW Policy
NASA NPD 2820.1 & NPR 7150.2

NASA SW Steering Board

CMMI Steering Group

4. Training, CMMI Appraisals & Career Development

5. SW Technology Infusion & Journal

6. Engineering, Assurance, & Safety Collaboration
NASA SW Process Asset Library
NASA SW Inventory
NASA Engineering Network (Software)

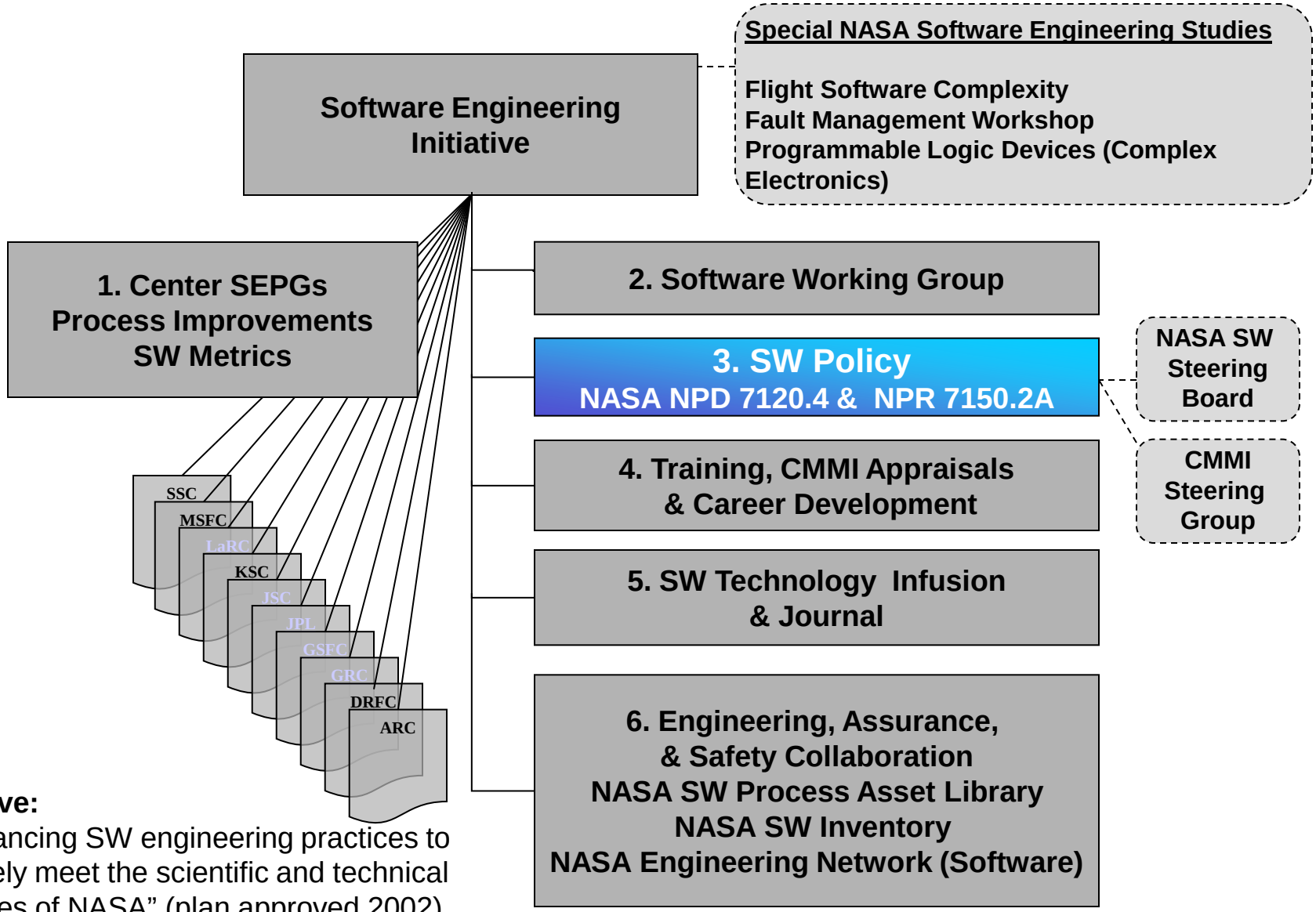
Objective:
"…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer

Chartered Roles of the Software Working Group (SWG)

- Function as an advisory group
- Recommend, draft as requested, review, and promote policies, standards, & best practices
- Recommend and provide technical support for special studies
- Focus, integrate, and promote innovation and the continuous improvement of NASA's software engineering processes
- Support and help guide the establishment of software process improvement programs at each Center
- Facilitate the transfer of software technology
- Coordinate NASA representation within agency, interagency, and international boards
- Provide information to improve communication on software issues
- Ad hoc activities as needed

Software Engineering Initiative



Objective:
"…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer

1.3.1 Higher Agency-Level Requirements

NPD 1000.0, NASA Governance and Strategic Management Handbook.
NPD 1000.3, The NASA Organization.
NPD 1000.5, Policy for NASA Acquisition.

1.3.2 Agency-Level Software Policies and Requirements

NPD 7120.4, NASA Engineering and Program/Project Management Policy

NPR 7120.5, NASA Space Flight Program and Project Management Requirements	NPR 7120.6, Lessons Learned Process	NPR 7120.7, NASA Information Technology and Institutional Infrastructure Program and Project Management Requirements	NPR 7120.8, NASA Research and Technology Program and Project Management Requirements	NPR 7123.1, NASA Systems Engineering Processes and Requirements	NPR 7150.2, NASA Software Engineering Requirements

1.3.3 Agency-Level Multi-Center and Product Line Requirements (non- software specific)

These NPDs and NPRs elaborate, tailor, and in some cases add requirements to the ones above to address the needs of major multi-Center projects, specific product lines, and specific focus areas.

1.3.4 NASA and Industry Software Standards and Guidebooks

NASA Preferred Industry Software Standards and Guidebooks and NASA Software-Related Standards and Guidebooks are required when invoked by an NPD, NPR, Center-Level Directive, contract clause, specification, or statement of work.

1.3.5 Center-Level Directives (related to software)

Center-Level Directives are developed by NASA Centers to document their local software policies, requirements, and procedures.

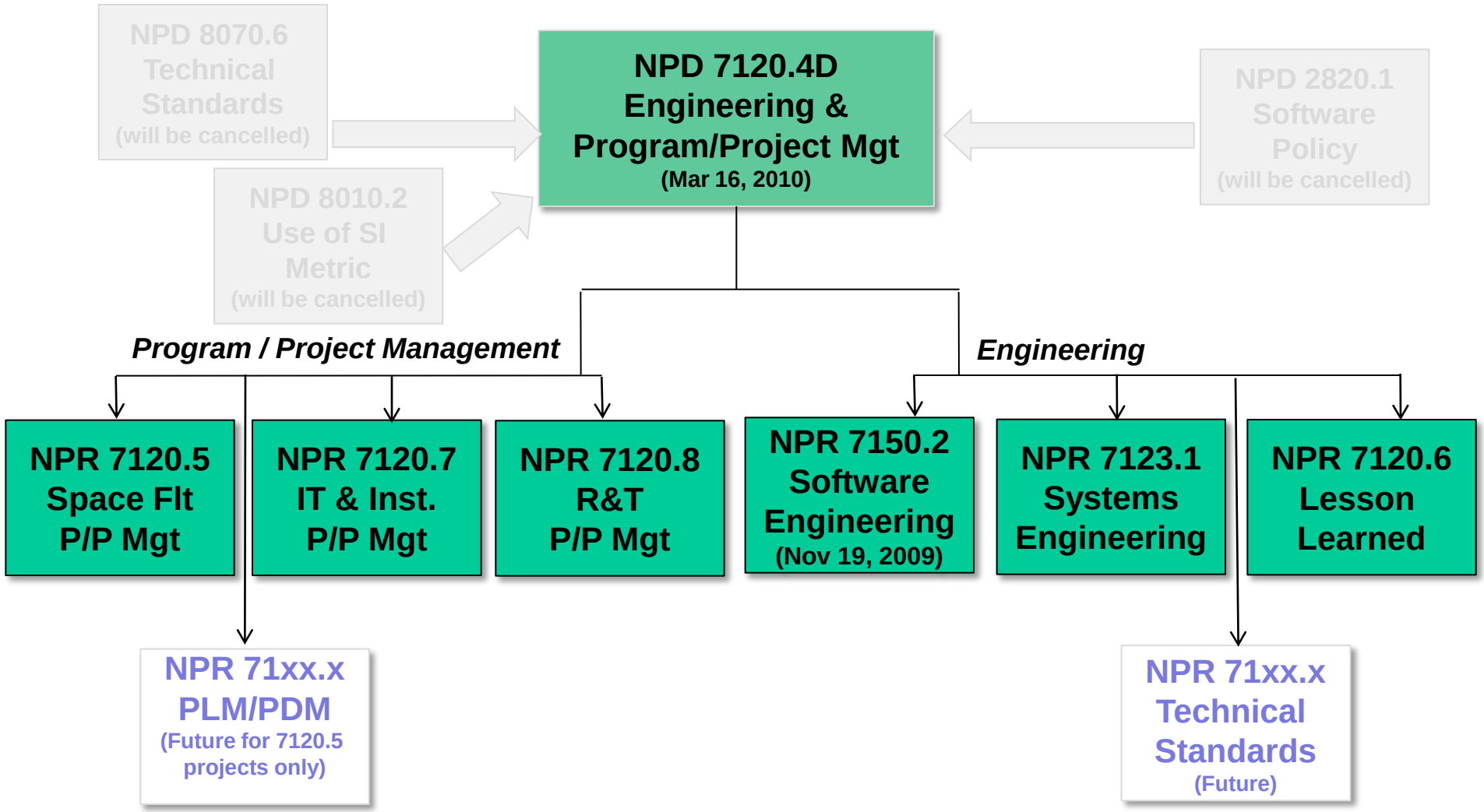
1.3.6 Government In-house Development

Government in-house software development policies and procedures to provide quality products and to fulfill the requirements passed down by a project.

1.3.7 Contractor and Subcontractor Development

Contractors and subcontractors develop in-house policies and procedures to provide quality products and to fulfill the requirements passed down through a contract by a customer.

Office of Chief Engineer Completed "Go To Architecture"



Note: NPD 7120.4D assumes NPD 2820.1 role in being the unique NASA NPD that covers software for all offices and organizations. NPD 7120.4 D is the *new parent NPD for NPR 2210.1, Release of NASA Softwar*

NPR 7150.2A TABLE OF CONTENTS

PREFACE

P.1 Purpose
P.2 Applicability and Scope
P.3 Authority
P.4 Applicable Documents
P.5 Measurement/Verification
P.6 Cancellation

CHAPTER 1. Introduction

1.1 Overview
1.2 Organizational Capabilities and Improvement
1.3 Hierarchy of NASA Software-Related Documents

CHAPTER 2. Software Management Requirements

2.1 Compliance with Laws, Policies, and Requirements
2.2 Software Life-Cycle Planning
2.3 Commercial, Government, Legacy/Heritage and Modified Off-The-Shelf Software
2.4 Software Verification and Validation
2.5 Project Formulation Requirements
2.6 Software Contract Requirements

CHAPTER 3. Software Engineering (Life-Cycle) Requirements

3.1 Software Requirements
3.2 Software Design
3.3 Software Implementation
3.4 Software Testing
3.5 Software Operations, Maintenance, and Retirement

CHAPTER 4. Supporting Software Life-Cycle Requirements

4.1 Software Configuration Management
4.2 Risk Management
4.3 Software Peer Reviews/Inspections
4.4 Software Measurement
4.5 Best Practices
4.6 Training

CHAPTER 5. Software Documentation Requirements

5.1 Software Plans
5.2 Software Requirements and Product Data
5.3 Software Reports

CHAPTER 6. Tailoring, Engineering Technical Authority, and Compliance Measurement

6.1 Tailoring of Requirements
6.2 Designation of Engineering Technical Authority(s)
6.3 Compliance

APPENDIX A. Definitions
APPENDIX B. Acronyms
APPENDIX C. References
APPENDIX D. Requirements Mapping Matrix
APPENDIX E. Software Classifications

Software is not all the same

flight software ⟺ *non-flight software*

engineering software ⟺ *general purpose software*

safety critical software ⟺ *non-safety critical software*

… and it shouldn't be treated the same!

NASA-wide Software Classification*

OCE	**Class A**	Space Flight Human Rated Software Systems
	Class B	Non-Human Space Rated Software Systems
	Class C	Mission Support Software & Facilities
	Class D	Analysis and Distribution Software
	Class E	Development Support Software
CIO	**Class F**	General Purpose Computing Software (Multi-Center or Multi-Program/Project)
	Class G	General Purpose Computing Software (Single Center or Project)
	Class H	General Purpose Desktop Software

Notes 1. "It is not uncommon for a project to contain multiple systems and subsystems having different software classes" (P.2.1)

2. Whether software is safety critical is an independent determination based on NASA-STD 8719.13

*** Established by NPR 7150.2**

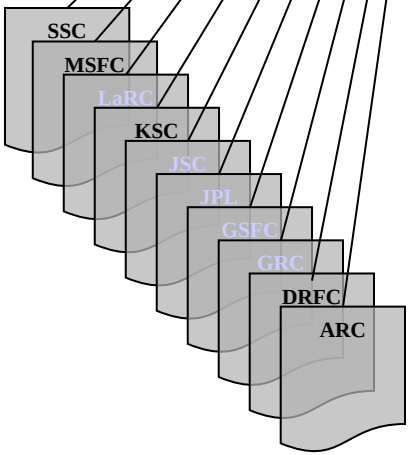
Software Engineering Initiative

Special NASA Software Engineering Studies

Flight Software Complexity
Fault Management Workshop
Programmable Logic Devices (Complex Electronics)

Software Engineering Initiative

1. Center SEPGs
Process Improvements
SW Metrics



2. Software Working Group

3. SW Policy
NASA NPD 2820.1 & NPR 7150.2

NASA SW Steering Board

CMMI Steering Group

4. Training, CMMI Appraisals & Career Development

5. SW Technology Infusion & Journal

6. Engineering, Assurance, & Safety Collaboration
NASA SW Process Asset Library
NASA SW Inventory
NASA Engineering Network (Software)

Objective:
"…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer

Software Engineering DACUM

Software Engineer DACUM – Phase 2 – Curriculum Plan

March 14th Baseline
Version 5.5

Proposed modifications for April 25th, 2006 Baseline

RUNNING STATUS ON SOFTWARE ENGINEERING DACUM CORRICULUM

Table 1: Status on Curriculum Classes to Date

Task ID by Phase II #	Course	PY [illegible]	C	Assignee	Vender	Action/Status	Due Date Close Date
2.2	101 SWE	E07		Pat / SG	?	-Description needs to be re-baselined to accommodate updates from Phase II report up through report version: v5.6-BASELINE-SW Engineering DACUM Curriculum.doc (updated SWE 101 emailed out by Pat to subgroup Tue, 04 Apr 2006 SWE101 V15.doc)	
2.3	101 SWE	E07		Z-TBD	?	-Hold on perusing finding class to fit description until after 201 and 301 complete (per April 06 F2F decision) –(or possibly do gap against [illegible] and this class)	
3.3	Software Requirements Development and Management	E07, 2		Cyrus/ Pat Tricia	APPEL / vendor CAI-POU 100 Requirement Development and Management; CAI-POU 100-T Project Team Training Requirement Development – Requirement management	-Work with vendor to fix gap on for Z123 "Use of state machines and discrete logic to model requirements" –Cyrus will look for appropriate class to can satisfy Z123. – Trisha will see if she can get "State Analysis for Software Engineering" [illegible]	Due FY06
3.4	Software Requirements Development and Management	E0 2					
3.4.2	Peer Reviews Inspection	E0 4					
3.4.3	Peer Reviews Inspection	E0 5					

Competency / Course Matrix	Introduction to Aerospace at NASA	SWE 101 Introduction to Software Engineering	Software Requirements Development and Management	Peer Reviews/ Inspections	SWE 110 Software Implementation	SWE 120 Software Testing	SWE 201 Software Engineering Processes	Software Configuration Management	Software Design	Software for Embedded Systems	Software Safety & Reliability	SWE 301 Software Management	Software and Process Metrics	Software Estimations	Software Process Improvement	Software Acquisition	Formal Methods for Software
Career Level (Early, Mid, Late)	EC	EC	EC	EC	EC	EC	MC	MC	MC	MC	MC	LC	LC	LC	LC	LC	LC
NASA Domain Knowledge																	
Knowledge of spacecraft and aeronautics systems, aerospace environment and architecture	•																
Knowledge of flight and ground systems, NASA Advanced Technology, Basic and Applied Research, and Institutional systems		•					•										
Knowledge of NPR 7150.2, NASA Core Standards, NASA Preferred Standards, and CMMI direction and guidance		•	•	•	•	•	•	•	•		•	•	•	•	•	•	
Software Acquisition and Procurement																	
Acquisition Processes, Contract Management and contractor performance		•					•					•				•	
Software Engineering Life Cycle Processes																	
Software Requirements Development		•	•														
Software Design		•			•	•			•	•							

Curriculum Plan for Software Engineering

Early Career Courses:

1. Introduction to Aerospace at NASA (IAN)
2. Software Engineering 101
3. Software Requirements Development and Management

3a. Peer Reviews/ Inspections (short separate class)

4. Software Implementation
5. Software Testing

Mid-Career Courses

6. Software Engineering 201

6a. Software Maintenance

7. Software Configuration Management
8. Software Design
9. Software for Embedded Systems
10. Software Safety & Reliability

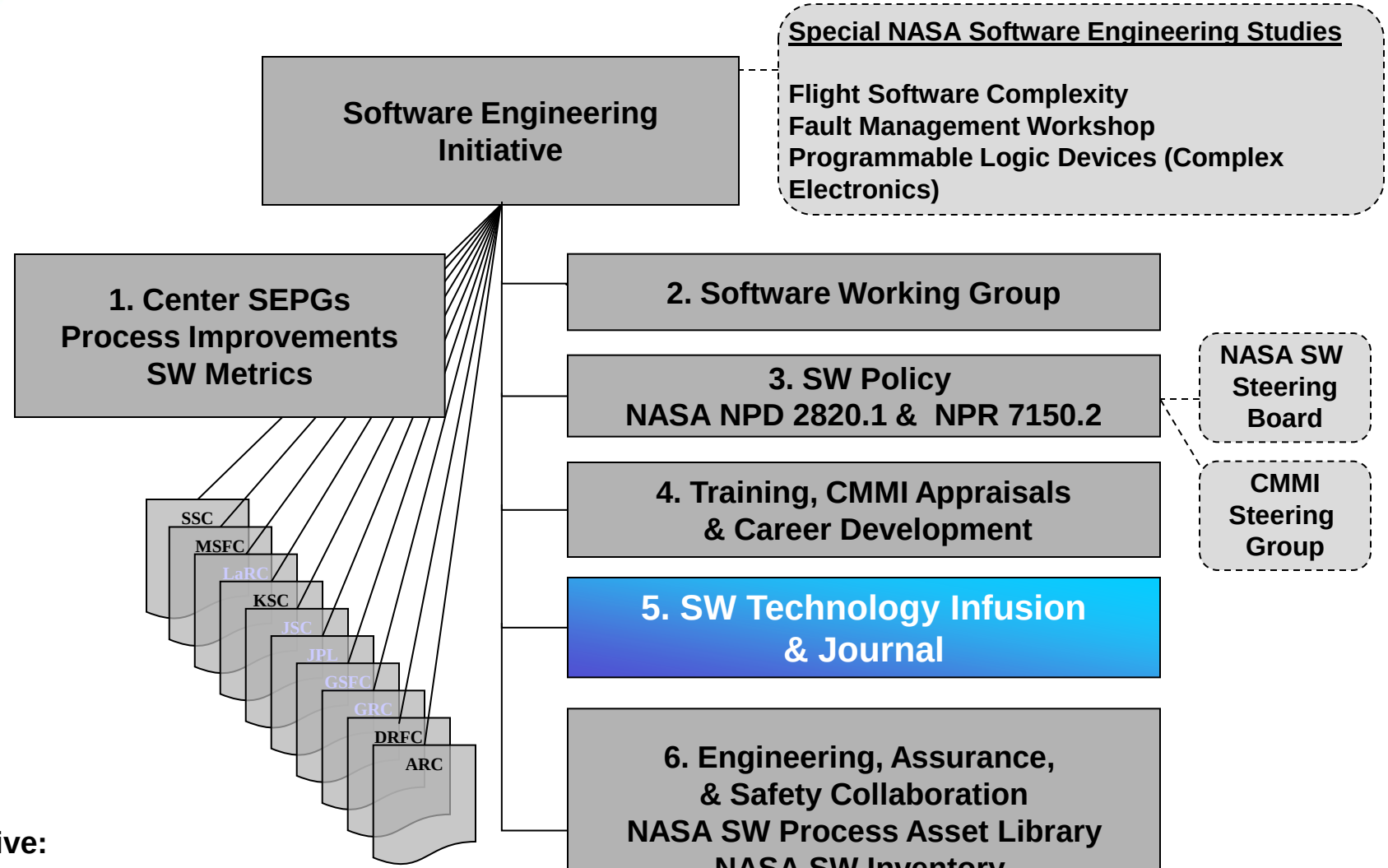
Later Career Courses

11. Software Engineering 301
12. Software and Process Metrics
13. Software Estimations
14. Software Process Improvement
15. Software Acquisition
16. Formal Methods for Software

Structured On-the-Job Learning

Informal On-the-Job Learning

Software Engineering Initiative



Objective:
"…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

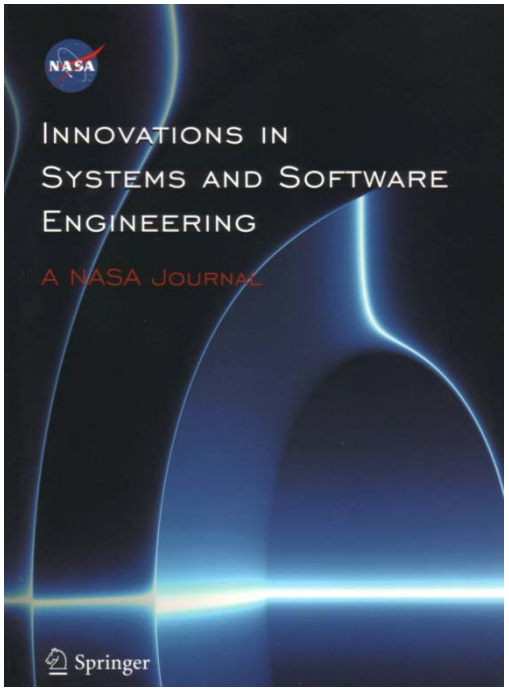
Sponsor: NASA Office of Chief Engineer

OSMA Software Assurance Research Program New Starts for FY10

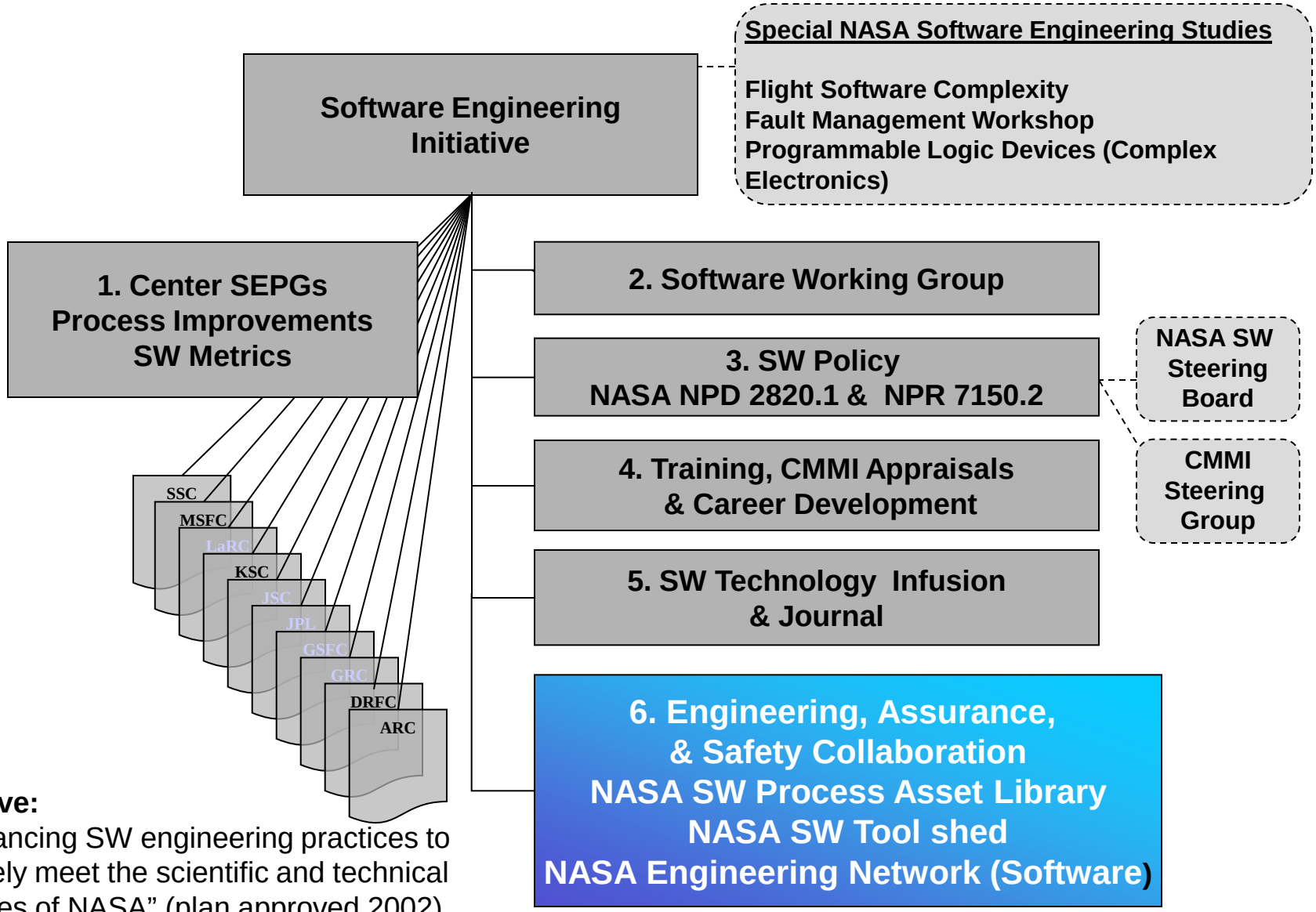
Initiative	PI	Center	The Work
Advanced Tools and Techniques for V&V of IVHM Systems	Johann Schuman	ARC	Analysis of complex IVHM systems require methods beyond traditional testing – will also add information and perspective to the on-going work on safety case
Architectural Analysis of Dynamically Reconfigurable Systems	Mikael Lindvall	GSFC	A follow-on to prior work, this initiative will be adjusting the work plan to support MSL and GMSEC
Command Process Modeling & Risk Analysis	Lelia Meshkat	JPL	The team will be developing tools and techniques to design and analyze robust command/operations process
Infusion of SDA for Automated Assessment of LADEE	Guille Del Carmin	ARC	Application of SDA (developed under NASA SBIR) on LADEE – collecting qualitative and quantitative information. This tool was previously tested by JSC MOD
Software Architecture Modeling and Assurance with AADL for the JPL SMAP Project	Katie Weiss	JPL	Application of the AADL approach of SMAP to be run in tandem with current processes
Toward Clear and Consistent Textual Requirements: An Application of Natural Language Processing Techniques	Allen Nikora	JPL	Developing approaches to support the automated discovery of ambiguous and inconsistent natural language requirements

Research & Technology Infusion Journal

- Encourage and stimulate advanced technology and research work that is relevant to NASA's missions
- Promote positive communication between the research and practicing professionals
- Supports the maturation of software and systems engineering related to aerospace applications
- Joint editorship between academic and government researchers
- Started in April 2005



Software Engineering Initiative



Objective:
"…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer

Approach on requirements related to the developmental aspects of safety critical software

2010 (NASA STD 8719.13 and STD 8739.8 updates)

NPR 7150.2.A, SW Engineering

Minimum SW Engineer Requirements base on SW Classifications A – H and software safety criticality

Generic Engineering Design Requirement for safety critical software systems

Specific Program and Project Requirements (w/Human Spaceflight track record)

Include program/project specific SW safety requirements as well as the implementation of NPR 7150.2A, Sec. 2.2.12

NASA SW Assurance and Safety Standards

Requirements for identifying and applying SW Assurance methods

Requirements to implement a systematic approach for software safety*

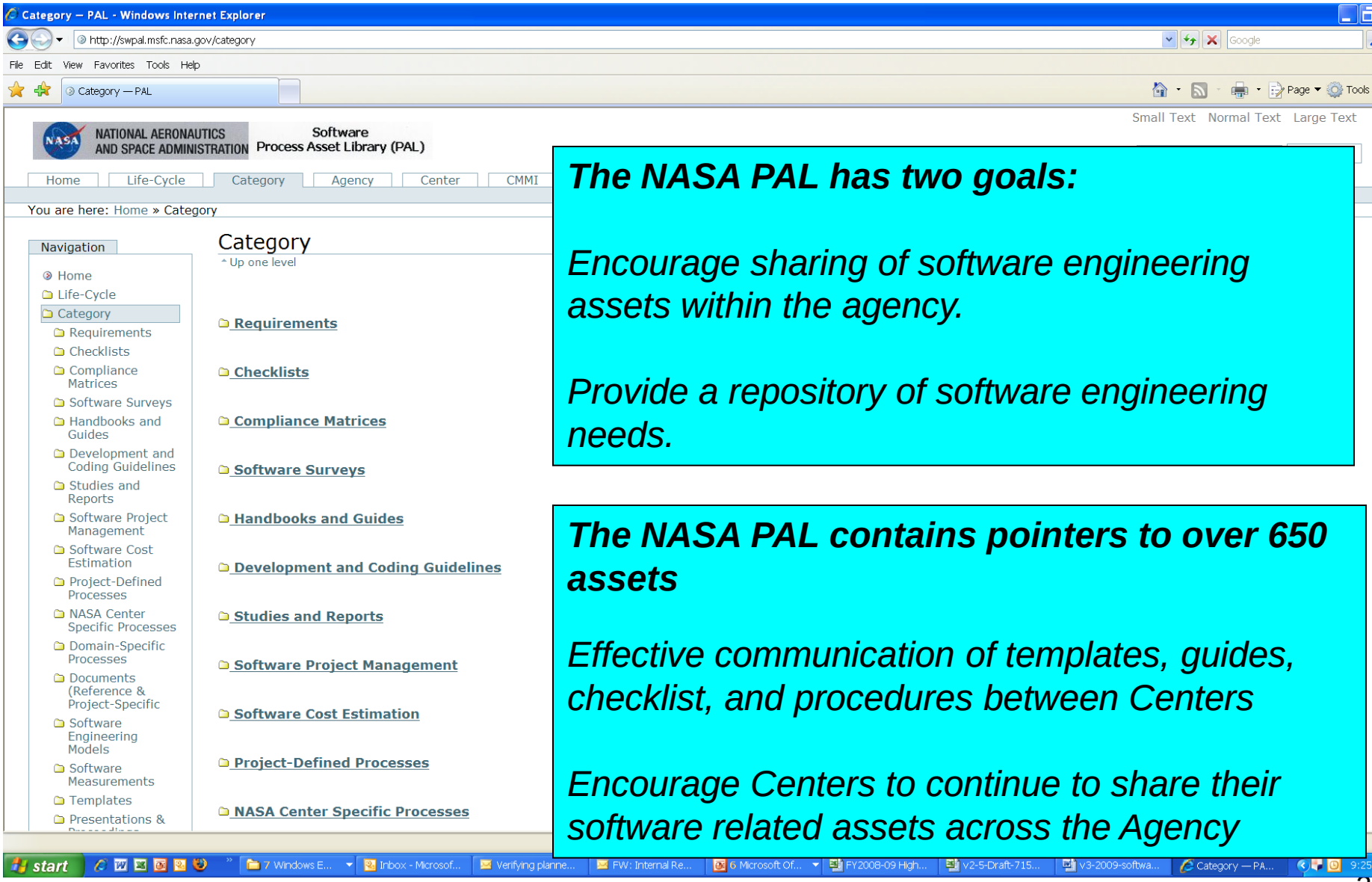
Set of SW safety requirements (and level of direction) beyond those found in NPR 7150.2 A

Solution: Harmonize Assurance & Safety Standards with NPR 7150.2A to resolve confusion over redundant aspects of documents

Note: The inclusion of some safety requirements in NPR 7150.2A does not relieve projects from complying with NASA STD 8719.13 and STD 8739.8

** Safety identification, assurance, risk & hazards analysis, FMEA,… remain in NASA STD 8719.13.*

NASA Process Asset Library



The NASA PAL has two goals:

Encourage sharing of software engineering assets within the agency.

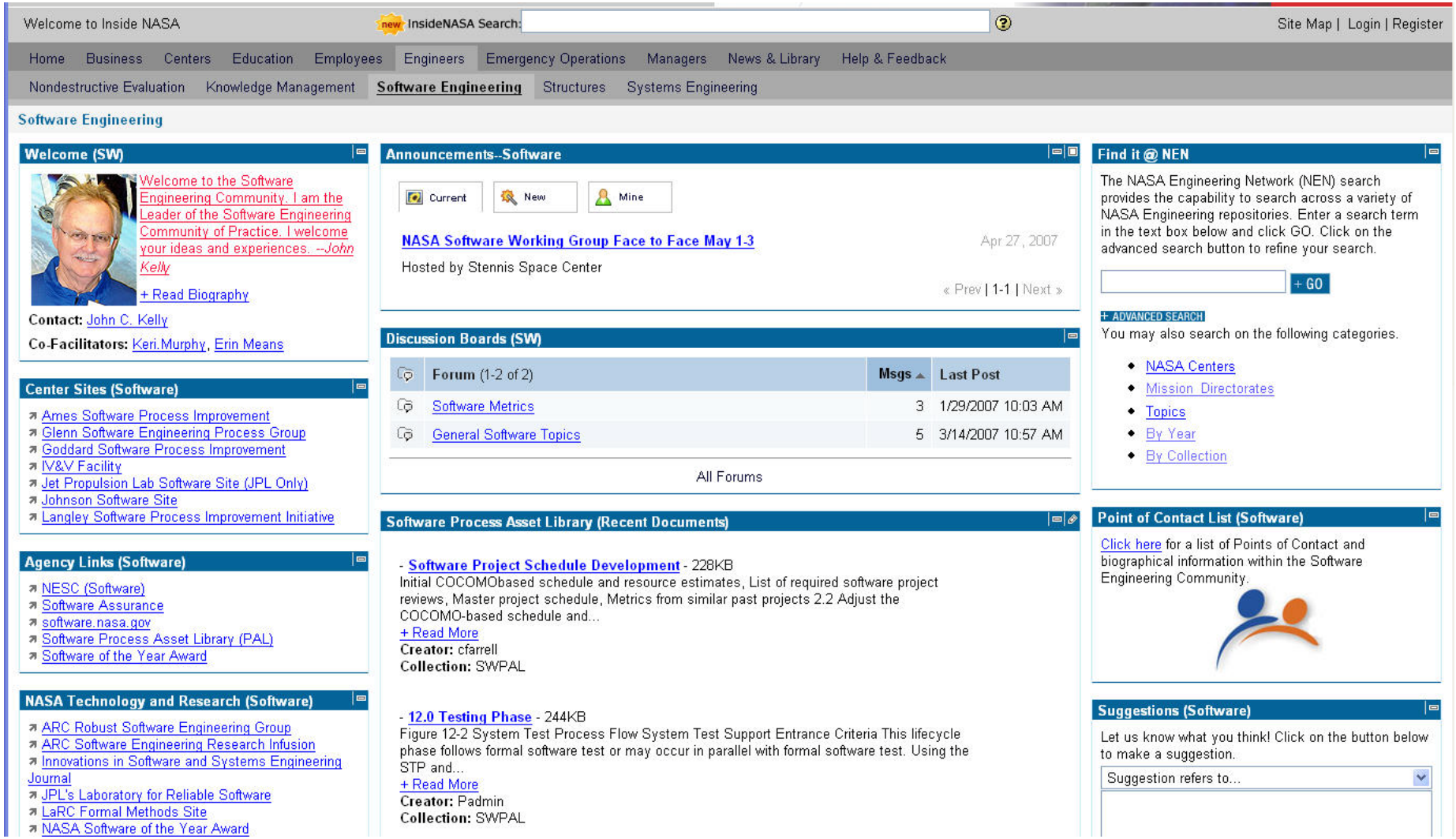
Provide a repository of software engineering needs.

The NASA PAL contains pointers to over 650 assets

Effective communication of templates, guides, checklist, and procedures between Centers

Encourage Centers to continue to share their software related assets across the Agency

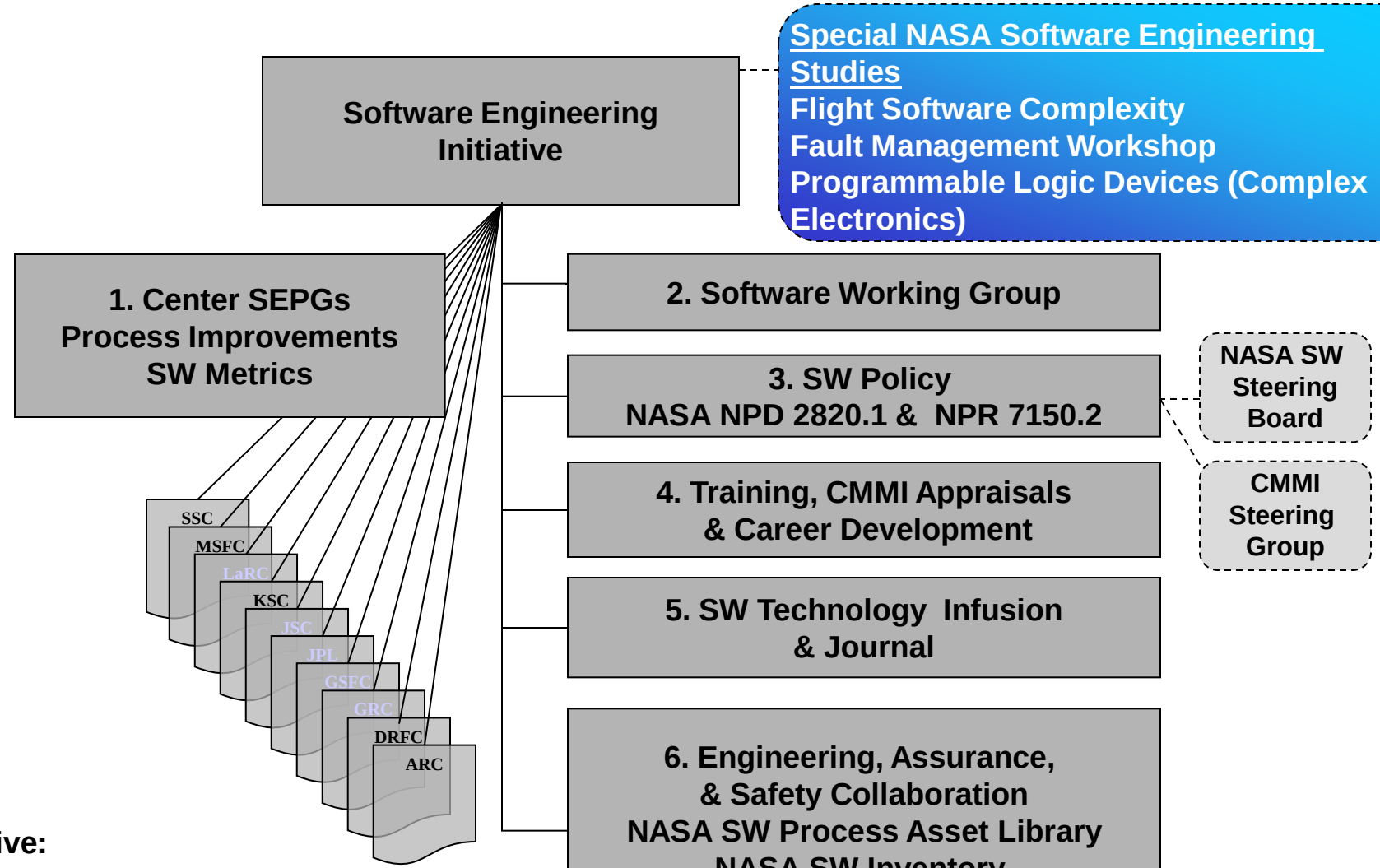
NASA Engineering Network
Software Engineering Portal



Software Tool Shed
Focus Area/Concept/Objective

- Deploy “in-house”, open source and COTS software tools (static analysis) on NASA mission software.
- Demonstrate the feasibility of an Application Service Provider model that provides missions with software tools and expertise.
- Taking a general approach to ASP; Language independent but a focus on Java with some C/C++ capability.
- Recognizing the need for “bug” reporting strategies for developers and managers.
- Building a relationship between NASA software engineering research, applied software engineering and mission software development (infusion and requirements).

Software Engineering Initiative

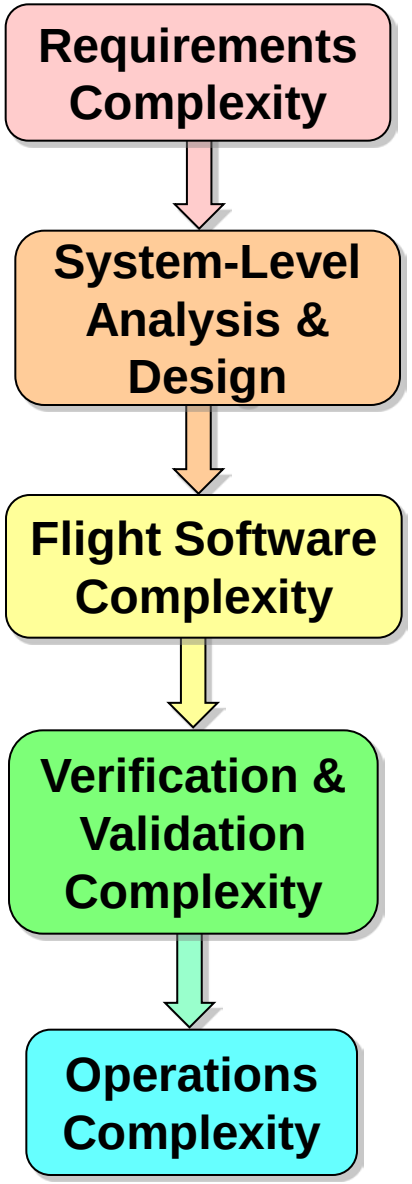


Objective:
"…advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer

NASA Study

Flight Software Complexity
Scope, Findings, Observations



- Challenging requirements raise downstream complexity (unavoidable)
- Lack of requirements rationale permit unnecessary requirements

- **Engineering trade studies not done: a missed opportunity**
- **Architectural thinking/review needed at level of systems and software**

- **Inadequate software architecture and lack of design patterns**
- **Coding guidelines help reduce defects and improve static analysis**
- **Descopes often shift complexity to operations**

- **Growth in testing complexity seen at all centers**
- **More software components and interactions to test**
- **COTS software is a mixed blessing**

- **Shortsighted FSW decisions make operations unnecessarily complex**
- **Numerous “operational workarounds” raise risk of command errors**

Programmable Logic Devices (Complex Electronics) NESC Problem Description

- Non descript discipline terms (“firmware”, “software” & “hardware”) have been used to describe a complicated device, which creates confusion
  - *Is an FPGA/ASIC containing a microprocessor function and associated code a hardware or software system?*
- No known single NASA-wide set of procedures, policy and/or guidelines exists for the design, development, test, and evaluation (DDT&E) of FPGA/ASICs for space flight applications.
- Historically, the application design’s operational speed and complexity has increased concurrently with the size of the circuitry decreasing
  - The single integrated circuit gives the appearance of minimal complexity
  - Past experience has uncovered undesirable features existing in designs

- This situation has all the ingredients of a pending accident
  - Complex design with critical functions **+** Difficultly in thoroughly testing all combinational logic modes **+** Varying DDT&E process **+** “It is only a chip” paradigm

Software Engineering Initiative

- Reduces risk of software failure - Increases mission safety
- More predictable software cost estimates and delivery schedules
- Smarter buyer of contracted out software
- More defects found and removed earlier
- Reduces duplication of efforts between projects
- Increases ability to meet the challenges of evolving software technology